Semi-divisive gene clustering and sparse principal components

Doyo Gragn[1] and Nickolay T. Trendafilov
Department of Mathematics and Statistics,
The Open University, Walton Hall, MK7 6AA, UK

Abstract

A new method for semi-divisive hierarchical clustering of variables (genes) is proposed. The method forms clusters of genes sequentially in two steps. First, the genes are ordered sequentially, either according to the highest sum of squared correlation, or based on the leading singular value of the already sorted genes and one of the unsorted ones. Then, the ordered genes are split in two parts such that the determinant of the correspondingly partitioned correlation matrix is maximized. The first group of genes becomes an output cluster, while the second one – input for another run of the sequential process. After the optimal clusters has been formed, sparse components can be constructed from the leading principal components in each cluster. The method is applied to a real gene expression data and the results compared with other existing approaches

Keywords: Gene expression; Hierarchical-clustering; Principal and sparse components

1 Introduction

Principal component analysis (PCA) (Jolliffe, 2002) is an efficient method for reducing the dimension of a high-dimensional data with minimal loss of information. The first few principal components (PCs) often account for the majority of the variation in the original data and are used for further analysis and interpretation. However, the interpretation of PCs is not always easy as each PC is a linear combination of all original variables. The problem is even worse with the gene expression data set due to the huge number of variables.

Several methods have been proposed for solving this problem using constrained optimization so that many of the PC loadings are forced to exact zeros (Jolliffe *et al.*, 2003; Zou *et al.*, 2006; Witten *et al.*, 2009). Such components with many exact-zero loadings are called sparse. Unlike the PCs, each sparse component is assumed to contain a 'cluster' of variables with nonzero loadings and the remaining

[1]For correspondence: D.Gragn@open.ac.uk

variables with exact-zero loadings. However, most of the existing methods produce sparse components with overlapping 'clusters' with respect to the nonzero-loading variables. Also, some of them produce components which are not sparse enough for interpretation. Such unwanted properties devaluate the approach.

Clustering could be one option to obtain sparse components with non-overlapping clusters of nonzero-loading variables and zero-loadings for the remaining ones. The aim of the cluster-based sparse component approach for gene expression data is to simplify the interpretation of PCs by computing sparse components from the non-overlapping clusters of genes. Thus, it involves both clustering and PCA. Once the variables are clustered, the nonzero loadings of a sparse component can be easily obtained by employing the standard PCA of the genes in each cluster. Each sparse component contains non-zero coefficient for all the genes in the cluster and zero coefficient for the remaining genes. As a result, the problem of finding best sparse components involves finding an optimal clustering method so that the cluster-based sparse components well approximate the PCs.

Cluster analysis is already known for its contribution in dimensionality-reduction in conjunction with other techniques such as PCA. For instance, Jolliffe (1972) employed clustering algorithm as a variable discarding technique in PCA. The purpose of his algorithm is to group the variables into clusters so that one variable or best subset of variables is selected from each cluster. The simple component analysis proposed by Rousson and Gasser (2004) uses clustering to identify b simple block components, which *overlap* with the rest $k - b$ simple difference components. For gene expression data, the purpose of gene-clustering might be to find genes that are potentially co-expressed. The gene-shaving method (Hastie *et al.*, 2000) is aimed at identifying subsets of genes with coherent expression patterns and large variation across samples. The gene-shaving approach also involves PCA. Many partitioning and hierarchical clustering methods have been proposed in literature for the gene expression data (Speed, 2003, Ch.4).

We propose a new clustering method, called the semi-divisive hierarchical clustering. The method forms clusters of genes sequentially in two steps. First, the genes are ordered sequentially according to either of the two criterion: the highest sum of squared correlation between the already sorted genes and one of the unsorted ones, or the leading singular value of the data matrix of the genes. Then, the ordered genes are partitioned into clusters based on the maximization of a determinant function involving the correlation matrices of the clusters of genes. In other words, the clusters are formed by partitioning the correlation or the data matrix of the variables at the "weakest-link" positions.

After the optimal clusters are formed, sparse principal components can be constructed from the leading principal components in each cluster. The idea of the cluster-based method for sparse component analysis is based on an optimal

clustering of variables in such a way that the correlations between the variables in one cluster and the variables in the other cluster are minimal.

The paper is organized as follows. The new technique for genes clustering is proposed in Section 2. The cluster-based sparse component method is outlined in Section 3. The developed techniques are applied to the gene expression of colon cancer data in Section 4, studied first by Alon *et al.* (1999) and publicly available at *http://microarray.princeton.edu/oncology/*. Section 5 summarizes the paper.

2 The semi-divisive method for gene clustering

Microarray gene-expression data is usually presented in a form of matrix with rows representing the different genes and columns representing the different cell lines (samples). The data matrix is characterized by having tens and hundreds of thousands of genes while the number of samples rarely exceeds a hundred. The information contained in such matrices is often overshadowed by the size of the data. One possible way of extracting the information is clustering.

Cluster analysis (McLachlan *et al.*, 2004; Seber, 2004) is an exploratory statistical technique concerned with grouping items or variables on the basis of similarities ("closeness") or dissimilarities ("distances"). The commonly available clustering criteria are often used to group samples (Friedman and Rubin, 1967), but there are also few methods used for grouping variables. The sample correlation coefficients or measures of association are the usual similarity measures for grouping variables. In some clustering applications, the absolute correlation coefficients are considered. Each cluster usually contains highly correlated variables, with each variable corresponding to one and only one cluster; i.e., the clusters are assumed to be non-overlapping with respect to the variables. In principle, no assumption should be made on the number of groups or group structure prior to analysis.

A number of approaches, both hierarchical and nonhierarchical, have already been proposed for clustering genes and/or samples for microarray gene expression data. An overview of the clustering methods is given by Tibshirani *et al.* (1999).

The clustering algorithm proposed in this Section is hierarchical. It forms clusters of genes sequentially in two steps: first ordering (sorting) the genes at each stage based on certain criterion, and then partitioning the ordered genes into clusters based on another criterion. At each stage of cluster formation, a type of *semi-divisive* hierarchical clustering method is used in which the ordered vector of genes are partitioned into two groups. The partitioning is made at the position of the "weakest-link" in the ordered genes where the 'gap' between the groups is maximum or the link is weak. The ordered genes are initially divided into two groups by maximizing a criterion involving the largest singular values of the data submatrices of the two groups. Then, the group with the larger criterion

value forms the first cluster, while the other group is subject to new ordering and partitioning. Thus, unlike the usual divisive hierarchical clustering, only one of the groups is allowed to divide at the next stage (hence the name semi-divisive).

2.1 Gene ordering

2.1.1 The leading singular value approach

Consider an $n \times p$ gene expression data matrix $\mathbf{X}$ with n samples (rows) and p genes (columns). The ordering of the genes is based on the *interlacing* property of its singular values (Horn and Johnson, 1985, Theorem 7.3.9(b)): If $\mathbf{X}_q$ is the matrix obtained by deleting the qth column of $\mathbf{X}$, then

$$d_1(\mathbf{X}) \geq d_1(\mathbf{X}_q) \geq d_2(\mathbf{X}) \geq d_2(\mathbf{X}_q) \geq \cdots \ldots \geq d_n(\mathbf{X}) \geq d_n(\mathbf{X}_q) \geq 0 \ , \qquad (1)$$

where $d_i(\mathbf{X}_q)$ denotes the ith singular value of $\mathbf{X}_q$.

The interlacing property suggests the following ordering procedure. First, find the pair of genes, for which the corresponding columns of $\mathbf{X}$, say $\{\mathbf{x}_{(1)}, \mathbf{x}_{(2)}\}$, has the largest singular value among all possible pairs of genes, i.e. the $n \times 2$ matrix $\mathbf{X}_{(2)} = [\mathbf{x}_{(1)}|\mathbf{x}_{(2)}]$ has the largest singular value among all possible $n \times 2$ submatrices of $\mathbf{X}$. Then, the next gene (with corresponding column $\mathbf{x}_{(3)}$) is chosen to be the one for which the augmented matrix $\mathbf{X}_{(3)} = [\mathbf{X}_{(2)}|\mathbf{x}_{(3)}]$ has the largest singular value among all remaining genes, and etc, until all genes are considered. Let $\mathbf{s}^{(q)}, q = 2, 3, ..., p$ denote the $q \times 1$ vector of indexes of the first q ordered genes. In general, the $(q+1)$th ordered gene, will be the one that maximizes:

$$d_1(\mathbf{X}_{(q+1)}) = d_1([\mathbf{X}_{(q)}|\mathbf{x}_j]) \ , \text{ over all } j \notin \mathbf{s}^{(q)} \ . \qquad (2)$$

2.1.2 The maximum sum of squared correlations approach

If the sample correlation matrix $\mathbf{R}$ is available, the genes can be sorted based on the sum of their squared correlation coefficients. Suppose, the highest squared correlation coefficient is r^2_{ij}, then form $\mathbf{s}^{(2)}$ with elements $s^{(2)}_1 = i$ and $s^{(2)}_2 = j$. Next, identify a gene with index k, that maximizes $r^2_{ik} + r^2_{jk}$ $(i \neq j \neq k)$ and set $s^{(3)}_3 = k$, i.e. $\mathbf{s}^{(3)} = [\mathbf{s}^{(2)}|s^{(3)}_3]$. Then, select the fourth gene, say $s^{(4)}_4 = l$, which maximizes $r^2_{il} + r^2_{jl} + r^2_{kl}$, and etc. The procedure continues in a similar way until all the genes are sorted. In general, the $(q+1)$th ordered gene, say m, will be the one that maximizes

$$\sum_{i=1}^{q} r^2_{s^{(q)}_i, m} \ , \text{ over all } m \notin \mathbf{s}^{(q)} \ . \qquad (3)$$

The vector of ordered variables is used in Section 2.2 to form a cluster. Then, the ordering algorithm repeats on the remaining un-clustered variables, and etc.

2.2 The partitioning criterion

Once the vector of indexes of the ordered genes $\mathbf{s} \equiv \mathbf{s}^{(p)}$ is found, the following criterion for partitioning is proposed. For the sake of simplicity, the formulation of the criterion is based on the correlation matrix of the ordered variables, though the computations latter involve the data matrix only. Denote by $\mathbf{R}$ the correlation matrix of the ordered genes and let $\mathbf{s}$ be partitioned into two vectors as $\mathbf{s} \equiv [\mathbf{s}_1 \mid \mathbf{s}_2]$ having k_1 and $p-k_1$ genes. Then $\mathbf{R}$ can be rewritten as the following block-matrix:

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{R}_{21} & \mathbf{R}_{22} \end{bmatrix},$$

where $\mathbf{R}_{ii}$ (i=1,2) is the correlation matrix of $\mathbf{s}_i$, and each element of the matrix $\mathbf{R}_{12} = \mathbf{R}_{21}^\top$ refers to the correlation coefficient between a gene from $\mathbf{s}_1$ and a gene from $\mathbf{s}_2$. The objective is to choose the partition in such a way that the following function (determinant) is maximized:

$$\begin{aligned} \delta(\mathbf{s}_1, \mathbf{s}_2) &= \det\left(\begin{bmatrix} \mathbf{v}_1^\top & \mathbf{0}^\top \\ \mathbf{0}^\top & \mathbf{v}_2^\top \end{bmatrix} \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{R}_{21} & \mathbf{R}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{v}_2 \end{bmatrix} \right) \\ &= \det\begin{pmatrix} \mathbf{v}_1^\top \mathbf{R}_{11} \mathbf{v}_1 & \mathbf{v}_1^\top \mathbf{R}_{12} \mathbf{v}_2 \\ \mathbf{v}_2^\top \mathbf{R}_{21} \mathbf{v}_1 & \mathbf{v}_2^\top \mathbf{R}_{22} \mathbf{v}_2 \end{pmatrix} \\ &= \left(\mathbf{v}_1^\top \mathbf{R}_{11} \mathbf{v}_1\right) \times \left(\mathbf{v}_2^\top \mathbf{R}_{22} \mathbf{v}_2\right) - \left(\mathbf{v}_1^\top \mathbf{R}_{12} \mathbf{v}_2\right)^2 . \end{aligned} \tag{4}$$

In other words, vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ should be found that maximize (4). Obviously, this function is maximized if $\mathbf{v}_1$ and $\mathbf{v}_2$ are the eigenvectors corresponding to the largest eigenvalues of $\mathbf{R}_{11}$ and $\mathbf{R}_{22}$, respectively, i.e.:

$$\max \delta(\mathbf{s}_1, \mathbf{s}_2) = d_1^2(\mathbf{R}_{11}) \times d_1^2(\mathbf{R}_{22}) - \left(\mathbf{v}_1^\top \mathbf{R}_{12} \mathbf{v}_2\right)^2 , \tag{5}$$

with $\mathbf{R}_{12} = \mathbf{0}_{k_1 \times (p-k_1)}$, if the genes in the two groups are uncorrelated.

As p is large, the eigenvalue-eigenvector pairs can be efficiently obtained from the SVD of the partitioned data matrices $\mathbf{X}_1 \in \mathbb{R}^{n \times k_1}$ and $\mathbf{X}_2 \in \mathbb{R}^{n \times (p-k_1)}$.

At the optimal solution of the first stage of the algorithm, $\mathbf{s}_1$ gives the first cluster of k_1 genes. To form the next cluster, the ordering and partitioning are repeated on the data matrix of the remaining $p - k_1$ vector of genes contained in $\mathbf{s}_2$. In general, the data matrix of $p - \sum_{i=0}^{k} k_i$ ordered genes is used at the ith stage of the partitioning procedure with $k_0 = 0$.

The whole procedure of gene-ordering and partitioning continue until the subgroup to be divided contains one gene or until one gets the required number of clusters (whichever comes first). If one is interested in a required number of clusters which exceeds the number obtained at the termination of the procedure, then it is possible to repeat the algorithm on one or more of the resulting clusters. At the extreme case, one can continue the procedure until each gene makes a cluster.

The quality of a cluster of q variables with indexes in $\mathbf{s}$ can be assessed as follows (Hastie *et al.*, 2000). Let

$$W_v = \frac{1}{n}\sum_{j=1}^{n}\left[\frac{1}{q}\sum_{\{i|\mathbf{x}_i\in\mathbf{s}\}}(x_{ij}-\bar{x}_j)^2\right]$$

and

$$B_v = \frac{1}{n}\sum_{j=1}^{n}(\bar{x}_j-\bar{x})^2$$

be the within-cluster and between-cluster variances of $\mathbf{s}$, respectively. Then the quality of the cluster can be measured by the variance ratio $[B_v/W_v]$ or the percent variance explained $[(B_v/T_v)\times 100]$ where $T_v = B_v + W_v$. The higher the percent variance explained, the tighter the coherence of the genes in the cluster.

The following iterative algorithm summarizes the semi-divisive clustering procedure:

1. Let $\mathbf{s}_0 = \{1, 2, ..., p\}$ contain the initial indexes of genes. Then, find the pair of indexes from $\mathbf{s}_0$, such that the $n \times 2$ matrix of the corresponding genes has the largest first singular value among all pairs from $\mathbf{s}_0$. Denote this pair by $\mathbf{s}_1$ and let $\mathbf{s}_2 \leftarrow \mathbf{s}\backslash\mathbf{s}_1$ ($\mathbf{s}$ without $\mathbf{s}_1$). Find $\delta(\mathbf{s}_1, \mathbf{s}_2)$.

2. Identify the gene from $\mathbf{s}_2$ which together with the two genes from $\mathbf{s}_1$ form a $n \times 3$ matrix with the largest singular value among all other genes from $\mathbf{s}_2$. Update $\mathbf{s}_1$ and $\mathbf{s}_2$ by removing this gene from $\mathbf{s}_2$ and inserting into $\mathbf{s}_1$. Find new $\delta(\mathbf{s}_1, \mathbf{s}_2)$, compare with the previous, and keep the largest.

3. Continue removing a gene from $\mathbf{s}_2$ and inserting it into $\mathbf{s}_1$, based on the maximization of (2), until $\mathbf{s}_2$ gets only one gene.

4. Identify the partition, say $\mathbf{s}_1^*$ and $\mathbf{s}_2^*$, that gives the largest value $\delta(\mathbf{s}_1^*, \mathbf{s}_2^*)$ of the criterion (5). Then $\mathbf{s}_1^*$ gives the first cluster of genes.

5. To get the next cluster, repeat the ordering and partitioning on the vector of genes $\mathbf{s}_2^*$ (i.e. $\mathbf{s}_0 \leftarrow \mathbf{s}_2^*$ and go to step 1).

6. Continue the algorithm until a required number of clusters is obtained or until each gene is included in one of the clusters.

The first three steps are based on the leading eigenvalue approach to gene ordering and can easily be adapted to the maximum sum of squared correlations approach.

3 Cluster-based sparse components

3.1 Motivation

The cluster-based approach for constructing sparse components is motivated by the specific form of the eigenvalue decomposition (EVD) of a block-diagonal correlation matrix. Let $\mathbf{R}$ be the following $p \times p$ block-diagonal correlation matrix:

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_{q_1} & \mathbf{0}_{q_1 \times q_2} & \cdots & \mathbf{0}_{q_1 \times q_k} \\ \mathbf{0}_{q_2 \times q_1} & \mathbf{R}_{q_2} & \cdots & \mathbf{0}_{q_2 \times q_k} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{q_k \times q_1} & \mathbf{0}_{q_k \times q_2} & \cdots & \mathbf{R}_{q_k} \end{bmatrix} , \tag{6}$$

where each block $\mathbf{R}_{q_i}$ is a $q_i \times q_i$ correlation matrix and $\sum_{i=1}^{k} q_i = p$. Then, the eigenvalues of $\mathbf{R}$ are solutions of the following equation:

$$f(d) = \det(\mathbf{R} - d^2 \mathbf{I}_p) = \prod_{i=1}^{k} \det(\mathbf{R}_{q_i} - d_i^2 \mathbf{I}_{q_i}) = 0 \ ,$$

i.e. the eigenvalues of $\mathbf{R}$ can be found by solving k smaller eigenvalue problems for $\mathbf{R}_{q_1}, ..., \mathbf{R}_{q_k}$ (Horn and Johnson, 1985, p.24). Let $\mathbf{R}_{q_i} = \mathbf{V}_{q_i} \mathbf{D}_{q_i}^2 \mathbf{V}_{q_i}^\top$ denote the EVD of $\mathbf{R}_{q_i}$. Then, after substitution in (6) one finds that

$$\mathbf{R} = \begin{bmatrix} \mathbf{V}_{q_1} \mathbf{D}_{q_1}^2 \mathbf{V}_{q_1}^\top & \mathbf{0}_{q_1 \times q_2} & \cdots & \mathbf{0}_{q_1 \times q_k} \\ \mathbf{0}_{q_2 \times q_1} & \mathbf{V}_{q_2} \mathbf{D}_{q_2}^2 \mathbf{V}_{q_2}^\top & \cdots & \mathbf{0}_{q_2 \times q_k} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{q_k \times q_1} & \mathbf{0}_{q_k \times q_2} & \cdots & \mathbf{V}_{q_k} \mathbf{D}_{q_k}^2 \mathbf{V}_{q_k}^\top \end{bmatrix} = \mathbf{V} \mathbf{D}^2 \mathbf{V}^\top \ , \tag{7}$$

where

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_{q_1} & \mathbf{0}_{q_1 \times 1} & \cdots & \mathbf{0}_{q_1 \times 1} \\ \mathbf{0}_{q_2 \times 1} & \mathbf{V}_{q_2} & \cdots & \mathbf{0}_{q_2 \times 1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{q_k \times 1} & \mathbf{0}_{q_k \times 1} & \cdots & \mathbf{V}_{q_k} \end{bmatrix} , \tag{8}$$

with $\mathbf{V}_{q_i}^\top \mathbf{V}_{q_i} = \mathbf{V}_{q_i} \mathbf{V}_{q_i}^\top = \mathbf{I}_{q_i}$ for each i, which implies $\mathbf{V}^\top \mathbf{V} = \mathbf{V} \mathbf{V}^\top = \mathbf{I}_p$, and

$$\mathbf{D}^2 = \begin{bmatrix} \mathbf{D}_{q_1}^2 & \mathbf{0}_{q_1 \times q_2} & \cdots & \mathbf{0}_{q_1 \times q_k} \\ \mathbf{0}_{q_2 \times q_1} & \mathbf{D}_{q_2}^2 & \cdots & \mathbf{0}_{q_2 \times q_k} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{q_k \times q_1} & \mathbf{0}_{q_k \times q_2} & \cdots & \mathbf{D}_{q_k}^2 \end{bmatrix} . \tag{9}$$

Thus, PCA of a block-diagonal correlation matrix results in a sparse loadings matrix (8). This feature was partially exploited by Rousson and Gasser (2004) for small p. In this paper, this feature is used to construct *orthogonal* sparse components.

3.2 Constructing cluster-based sparse components

Assume that the variables (genes) are already grouped into k clusters and that the jth cluster is composed by q_j variables, for $j = 1, \ldots, k$ and $\sum_{j=1}^{k} q_j = p$. Let $\boldsymbol{\omega}_j$ be the $q_j \times 1$ vector containing the indexes of the original variables clustered into the jth cluster in ascending order, i.e. $\omega_{1,j} \leq \omega_{2,j} \leq ... \leq \omega_{q_j,j}$. Define the following $p \times q_j$ indicator matrices $\mathbf{G}_j$, for $j = 1, \ldots, k$: $\mathbf{G}_j$ has 1 at its position $(\omega_{l,j}, l)$ for $l = 1, \ldots, q_j$ and 0 otherwise. Then $\mathbf{X}_j = \mathbf{X}\mathbf{G}_j$ is the $n \times q_j$ data submatrix corresponding to the jth cluster and let $\mathbf{X}_j = \mathbf{U}_j\mathbf{D}_j\mathbf{V}_j$ be its singular value decomposition (SVD). Denote by $\mathbf{v}_j$ the singular vector of $\mathbf{V}_j$ corresponding to the largest singular value in $\mathbf{D}_j$. Then, the $p \times k$ matrix of cluster-based sparse component loadings $\mathbf{B}$ is formed as follows:

$$\mathbf{B} = [\mathbf{G}_1\mathbf{v}_1|\mathbf{G}_2\mathbf{v}_2|...|\mathbf{G}_k\mathbf{v}_k] \ .$$

The goodness-of-fit for the sparse components can be measured using the percent of variances explained by the components. This measure is based on the cumulative percentage of the variances (eigenvalues) explained by the sparse components in relation to the total sum of the eigenvalues of the data matrix. As the sparse components are correlated to each other, the adjusted variances (Zou *et al.*, 2006) are used as a better measure of goodness-of-fit.

3.3 Number of sparse components

The number of cluster-based sparse components depends on the number of clusters. But, there is no hard rule for choosing the number of clusters in cluster analysis, though there are some suggestions (Seber, 2004, p.388). There are few attempts to determine the number of clusters in the microarray gene expression data (McLachlan *et al.*, 2004, Section 4.12). The sparsity level is also affected by the number of components. Unlike the constrained sparse techniques (such as the LASSO-based methods), which constrain the number of non-zero loadings in a sparse component by introducing a tuning parameter, the proposed method controls the sparsity level by the number of clusters. The higher the number of clusters, the sparser the components.

The new clustering algorithm proposed in Section 2 finds k clusters, which number is unknown *a priori* and is a result of a particular optimal ordering/partitioning process. This implies that the number of sparse components should not be always prescribed in advance, say based on the scree plot of the original data.

It is also possible to constrain the number of clusters to a required number k' where $k' \leq k$. This number k' is supposed to govern the dimension and the sparsity of the components. However, if one is interested in k' clusters where $k < k' \leq p$, then it is possible to repeat the algorithm on one or more of the

clusters themselves. For this purpose, the next possible cluster to be partitioned into two further clusters could be the one which gives the "maximum weakest-link" between the partitions.

4 Application

The gene expression measurements considered by Alon *et al.* (1999) and publicly available from http://microarray.princeton.edu/oncology/ are studied in this Section. The data matrix consists of 2000 genes in 62 samples, 40 tumor and 22 normal colon tissue samples. Some of the genes in the data set are duplicated: there are more than one different expression sequences. As such genes are highly correlated, only one of the sequences, which has the largest standard deviation, is considered. This procedure reduces the number of unique genes to 1909. Thus, the final data is a mean-centered and scaled matrix of the (natural) logarithm of $p = 1909$ genes in $n = 62$ samples.

4.1 Result

The maximum sum of squared correlation coefficients approach (Section 2.1.2) is used to sort the genes. The semi-divisive clustering algorithm results in 12 clusters, with sizes ranging from 1 (for the last cluster) to 663 (for the second cluster). The cluster sizes are given in Table 1.

To check whether these clusters are genuine with respect to the expression patterns, two heat maps are plotted: one for the genes before clustering, and another one – for the genes in the first five clusters (containing 95.6% of the total genes). The two heat maps are given in Figure 1. The rows of the heat map represent the genes and the columns represent the samples (with columns 1–40 for the tumor and columns 41–62 for the normal samples). The genes in the first heat map are given in the alphabetical order of their code. The genes in each cluster for the second heat map are given in the order they join the cluster. The heat maps are plotted using a heat map builder freely available at *http://ashleylab.stanford.edu/tools_scripts.html*.

The five clusters (viewed horizontally) are clearly visible from the heat map of the clustered genes (right plot, Figure 1) compared to that of the unclustered genes (left plot, Figure 1). The result could be a simple and visual proof that the method has performed well in clustering the genes. The quality of the five clusters is assessed based on the measure involving the within-cluster and between-cluster variances as given in Section 2. This gives the following explained percentages of variances: 66.9%, 57.2%, 60.8%, 65.6%, and 54.5%.

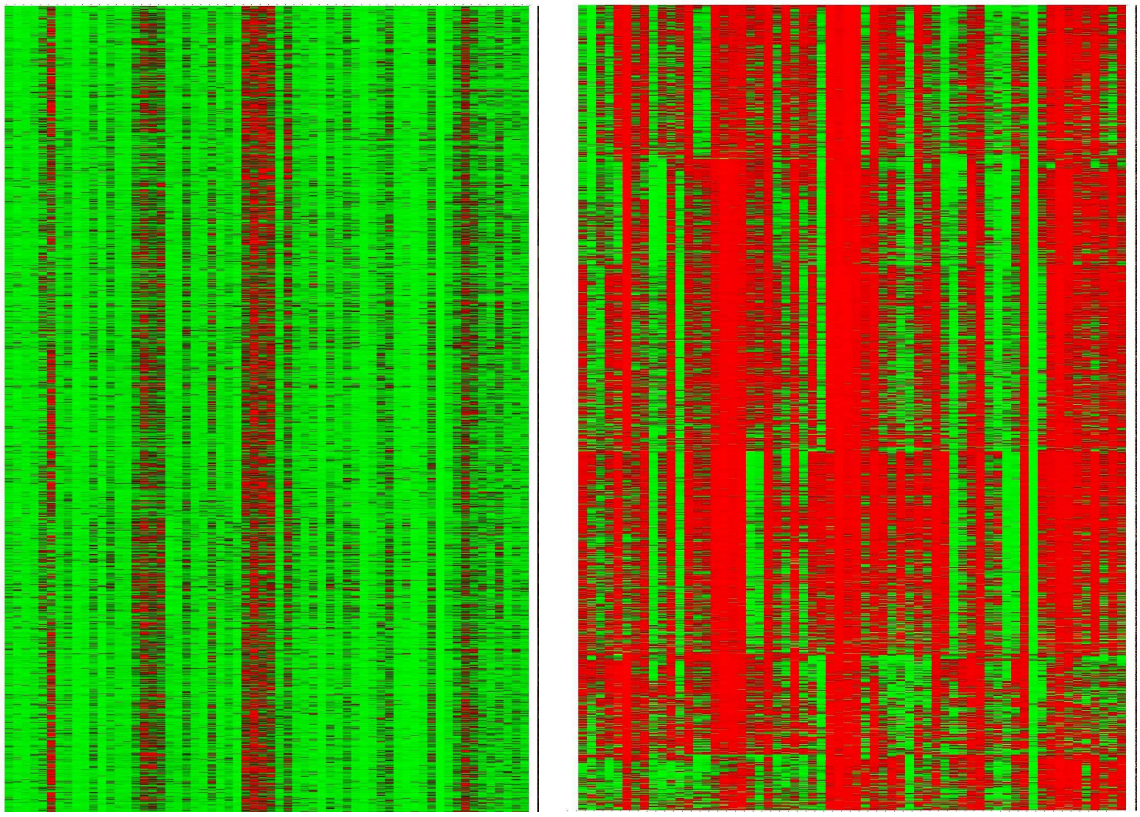

Figure 1: *(Left) Heat map of all the genes in the data before clustering; (Right) Heat map of the genes in the first five clusters (ordered from top to bottom), Alon data*

The sparse components were computed corresponding to the resulting clusters. The number of nonzero-loadings (equal to the cluster sizes) is given in Table 1, including the unadjusted and adjusted variances and cumulative variances explained by the sparse components. Regarding the computing speed, the algorithm took 14.5 minutes to give the solution on an Intel(R) Pentium 4 computer with 3.2GHz CPU and .99 GB of Ram.

4.2 Comparison of clusters from the semi-divisive and gene shaving methods

McLachlan *et al.* (2004) used the colon data of Alon *et al.* (1999) to demonstrate the (unsupervised) gene shaving method (Hastie *et al.*, 2000). For the sake of comparison, we consider the genes in the four gene-shaving clusters depicted on

Table 1: *Number of nonzero (#NZ) loadings per sparse component and percentage of variance (V) & cumulative variance (CV) explained, Alon data*

Comp	1	2	3	4	5	6	7	8	9	10	11	12
#NZ	349	663	460	228	126	50	10	7	11	2	2	1
V	12.94	22.64	15.45	5.36	3.12	1.03	.28	.17	.22	.08	.07	.05
CV	12.94	35.58	51.03	56.39	59.51	60.54	60.82	60.99	61.21	61.29	61.36	61.41
V_{adj}	12.94	10.22	7.74	1.67	.98	.14	.15	.04	.07	.03	.05	.03
CV_{adj}	12.94	23.16	30.90	32.57	33.53	33.67	33.82	33.86	33.93	33.96	34.01	34.04

p.181 of McLachlan *et al.* (2004). Table 2 relates the genes in the four gene-shaving clusters to those obtained by the proposed here semi-divisive clustering method. The table shows that all the genes in the first gene-shaving cluster fall into the first semi-divisive cluster. Similarly, all the genes in the second gene-shaving cluster fall into the third semi-divisive cluster. On the other hand, almost all the genes in both the third and the fourth gene-shaving clusters (except three genes) are grouped into the second semi-divisive cluster.

To see if the semi-divisive clustering method can identify the third and the fourth gene-shaving clusters, we applied the ordering and partitioning procedures to the 663 genes in the second semi-divisive cluster and grouped it into two further clusters. The result showed that, all the genes in the fourth gene-shaving cluster fall into one of the new semi-divisive clusters while all but three genes in the third gene-shaving cluster (among those already in the second sparse component in the first stage) fall into another cluster. At this second stage, the two new clusters contain 239 and 424 genes.

In the semi-divisive method, the number of clusters as well as the cluster sizes are fixed by the algorithm itself. It usually results in small number of clusters, each with large number of genes. This is partly due to the semi-divisiveness property of the algorithm: only one side is partitioned on the next phase. This is also related to the objective of the algorithm in that the resulting clusters are designed to be used for constructing sparse components with maximal explained variance. The above second-round clustering step is conducted just for comparison purpose and is not required with respect to our objective. However, if one is interested solely on the clustering, the procedure can be repeated on the resulting clusters so that the number of clusters increases (and hence the cluster size decreases).

In contrast, the gene-shaving method requires fixing the number of clusters *a priori* and estimates the optimal cluster size using the 'gap statistic' (Tibshirani *et al.*, 2001). It is designed to extract small clusters of genes that vary as much as possible across the samples. Unlike the semi-divisive method, genes in the

gene-shaving method may belong to more than one cluster.

Table 2: *Cluster membership in the sparse component (SC) of the genes clustered by gene-shaving (GS), Alon data*

GS #1	SC #	GS #2	SC #	GS #3	SC #	GS #4	SC #
L02426	1	R34876	3	U21914	2	U27143	2
M26697	1	T57686	3	R15814	2	R49231	2
T51023	1	T60437	3	D26018	2	R43913	2
R43914	1	T57468	3	R33367	2	X72727	2
M84326	1	X12466	3	D14689	2	R22779	2
M88279	1	M29065	3	L10413	2	L19437	2
M22382	1	T52642	3	U14588	2	T69748	2
M14200	1	H24030	3	R53936	2	T70595	4
T69446	1	T56244	3	D26067	2	T92259	2
T93589	1	H05899	3	D13641	2	H88250	2
T84049	1	T63591	3	R09468	2	X68194	2
T40674	1	H69869	3	R71585	2	H09719	2
R60859	1	T65758	3	D21260	2	D14043	2
H89087	1	U02493	3	D13627	2	D17400	2
R37428	1	M21339	3	U18062	2	H38185	2
R16156	1			L10911	2	H42127	2
D00761	1			R27813	2	X87838	2
				M90104	2	L19437	2
				X01060	2	D15057	2
				R50864	1	U20998	2
				X16135	1		

4.3 Comparison with sparse components from other methods

The proposed cluster-based sparse components method is compared with the sparse principal component (SPC) method (Witten *et al.*, 2009) and with the sparse principal component analysis (SPCA) (Zou *et al.*, 2006) by analyzing the Alon colon data. The comparison involve the level of sparsity and the percent variance explained.

The `SPC` function in `R` (Witten *et al.*, 2009) requires as one of its arguments the sum of absolute values (`sumabsv`) of loadings in a sparse component. This value is assumed to measure the level of sparsity and is set by the user. For comparison, two different values for `sumabsv` are used. The first is the maximum of the sum of the absolute values of the SC loadings from our method, and the second is the average of the sum of the absolute values of the SC loadings. For

the gene expression data under consideration, these values are found to be 25.6861 and 9.2921, respectively, with 12 SCs. The top two plots in Figure 2 are based on `sumabsv` = 25.6861, and the bottom two – on `sumabsv` = 9.2921. The SPCA results are adjusted as explained in (Witten *et al.*, 2009).

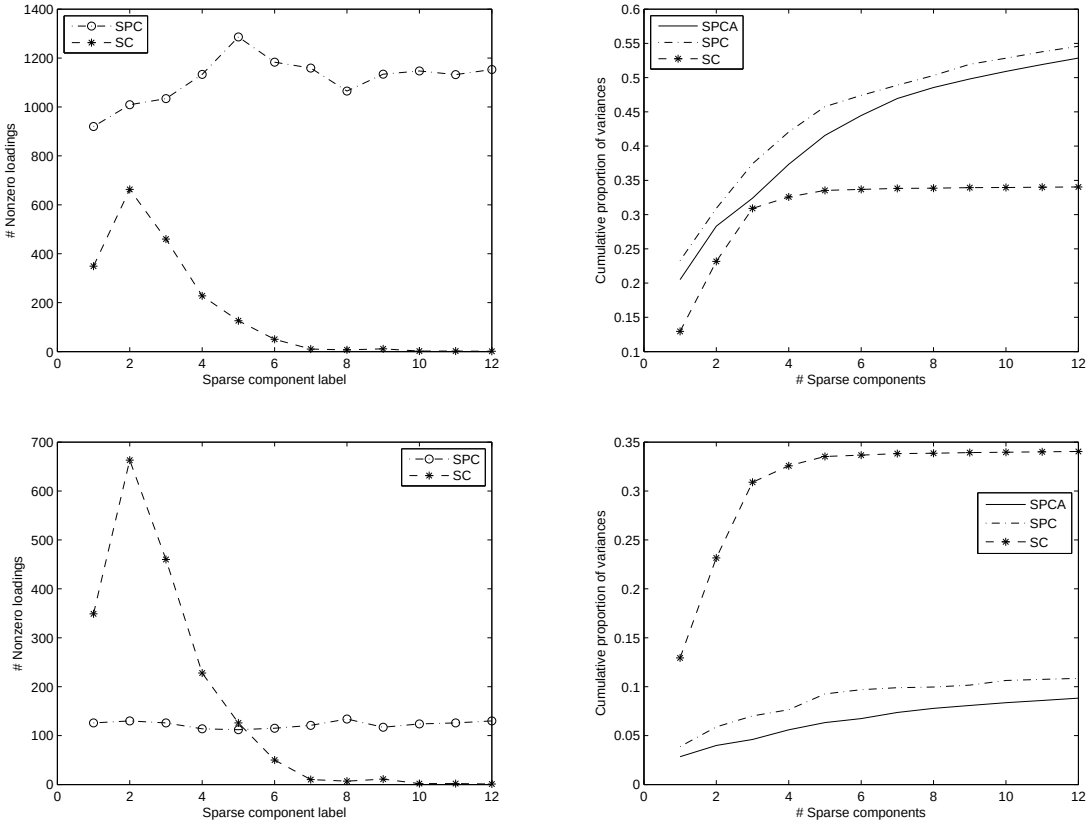

Figure 2: *Comparison of SC, SPC and SPCA with respect to sparsity (Left) and the cumulative proportion of adjusted variance explained (Right). The two upper plots are based on* `sumabsv` = 25.6861 *and the two lower plots are based on* `sumabsv` = 9.2921 *for SPC.*

The methods give varying results for the sparsity level and the cumulative proportion of variances explained. The SPC method highly depends on the value of `sumabsv`: higher values result in less sparse components (but with higher cumulative variance explained), and vice versa. The level of sparsity for the SC is generally decreasing with the number of sparse components, while the SPC gives almost similar level across the components. The cumulative variances explained by the sparse components are also affected by the level of sparsity.

5 Summary

Principal component analysis is a known and efficient technique for reducing the dimension of a high-dimensional data. However, the principal components, as a linear combination of all the original variables, are not easily interpretable. There are many approaches suggested to simplify the PC interpretation, of which sparsifying the component loadings is one possible option. The sparse components might not still be sparse enough or the different sparse components share few of the variables with nonzero loadings. These behaviors may not be attractive for the components to be more interpretable. Decision on the number of components is another problem, even when working with the PCA.

The cluster-based sparse components are designed to be computed from an optimally clustered variables. The procedure involves two major tasks: clustering the genes and application of PCA to each cluster to form the sparse components. Thus, each component includes a unique set of nonzero-loading genes, and no sparse component is overlapping with respect to the genes. The number of sparse components depends on the number of clusters.

A new semi-divisive hierarchical clustering method is proposed, which results in an optimal number of clusters not known in advance. Each stage of the method involves two steps: ordering (sorting) the genes based on certain criterion and partitioning the ordered genes based on another criterion. The application of the method to a real gene expression data shows that the semi-divisive technique is a fast and powerful method for gene expression clustering and interpretation.

References

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., and Levine, A. J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci. USA*, **96**, 6745–6750.

Friedman, H. P. and Rubin, J. (1967). On some invariant criteria for grouping data. *J. Am. Statist. Ass.*, **62**, 1159–1178.

Hastie, T., Tibshirani, R., Eisen, M. B., Alizadeh, A., Levy, R., Staudt, L., Chan, W. C., Botstein, D., and Brown, P. (2000). 'Gene shaving' as a method for identifying distinct sets of genes with similar expression patterns. *Genome Biology*, **1**, research0003.1–0003.21.

Horn, R. A. and Johnson, C. A. (1985). *Matrix Analysis*. Cambridge University Press, Cambridge.

Jolliffe, I. T. (1972). Discarding variables in a principal component analysis i: Artificial data. *Applied statistics*, **21**, 160–173.

Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer-verlag, New York.

Jolliffe, I. T., Trendafilov, N. T., and Uddin, M. (2003). A modified principal component technique based on the lasso. *Journal of Computational and Graphical Statistics*, **12**, 531–547.

McLachlan, G. J., Do, K.-A., and Ambroise, C. (2004). *Analyzing Microarray Gene Expression Data*. Wiley, New Jersey.

Rousson, V. and Gasser, T. (2004). Simple component analysis. *Applied Statistics*, **53**, 539–555.

Seber, G. A. F. (2004). *Multivariate Observations*. Wiley, New Jersey.

Speed, T., editor (2003). *Statistical Analysis of Gene Expression Microarray Data*. Chapman & Hall, New York.

Tibshirani, R., Hastie, T., Eisen, M., Ross, D., Botstein, D., and Brown, P. (1999). Clustering methods for the analysis of dna microarray data. Technical report. Stanford: Department of Statistics, Stanford University.

Tibshirani, R., Walther, G., and Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *J. Royal Stat. Soc. B*, **63**, 411–423.

Witten, D. M., Tibshirani, R., and Hastie, T. (2009). A penalized matrix decomposition, with applications to sparse principal components and canonical correlation. *Biostatistics*, **10**, 515–534.

Zou, H., Hastie, T., and Tibshirani, R. (2006). Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, **15**, 265–286.